

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: August 26, 2021

B. Decraene  
Orange  
R. Raszuk  
NTT Network Innovations  
Z. Li  
C. Li  
Huawei Technologies  
February 22, 2021

**SRv6 vSID: Network Programming extension for variable length SIDs  
draft-decraene-spring-srv6-vlsid-05**

**Abstract**

This document proposes an extension to Segment Routing IPv6 (SRv6) Network Programming to allow for SRv6 Segment Identifier (SID) of smaller variable length. The use of smaller SRv6 SID reduces the size the SRv6 Header (SRH). This reduces the overhead for both the traffic volume and the network processor. It is a straightforward extension to the SRv6 Network Programming model and its SRH encapsulation.

**Status of This Memo**

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 26, 2021.

**Copyright Notice**

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction</a>	<a href="#">2</a>
<a href="#">1.1.</a>	<a href="#">Requirements Language</a>	<a href="#">3</a>
<a href="#">2.</a>	<a href="#">Overview</a>	<a href="#">3</a>
<a href="#">3.</a>	<a href="#">SRv6 Variable Length SIDs</a>	<a href="#">4</a>
<a href="#">3.1.</a>	<a href="#">Encoding vSIDs in the SRH</a>	<a href="#">5</a>
<a href="#">3.2.</a>	<a href="#">SRv6 vSIDs Network Programming</a>	<a href="#">7</a>
<a href="#">4.</a>	<a href="#">Benefits</a>	<a href="#">7</a>
<a href="#">5.</a>	<a href="#">Illustrations</a>	<a href="#">8</a>
<a href="#">5.1.</a>	<a href="#">Global vSIDs</a>	<a href="#">8</a>
<a href="#">5.2.</a>	<a href="#">Local vSIDs</a>	<a href="#">9</a>
<a href="#">6.</a>	<a href="#">Optional extensions</a>	<a href="#">10</a>
<a href="#">6.1.</a>	<a href="#">A node requiring a larger vSIDs length.</a>	<a href="#">10</a>
<a href="#">6.2.</a>	<a href="#">Inter Routing Domains with the End.XvPS behavior</a>	<a href="#">12</a>
<a href="#">6.3.</a>	<a href="#">NSIDs flavor</a>	<a href="#">13</a>
<a href="#">7.</a>	<a href="#">Security Considerations</a>	<a href="#">14</a>
<a href="#">8.</a>	<a href="#">Acknowledgements</a>	<a href="#">14</a>
<a href="#">9.</a>	<a href="#">Contributors</a>	<a href="#">14</a>
<a href="#">10.</a>	<a href="#">Changes / Authors Notes</a>	<a href="#">14</a>
<a href="#">11.</a>	<a href="#">References</a>	<a href="#">15</a>
<a href="#">11.1.</a>	<a href="#">Normative References</a>	<a href="#">15</a>
<a href="#">11.2.</a>	<a href="#">Informative References</a>	<a href="#">16</a>
	<a href="#">Authors' Addresses</a>	<a href="#">17</a>

## [1.](#) Introduction

The Segment Routing (SR) architecture is defined [RFC 8402](#) [[RFC8402](#)].

IPv6 Segment Routing Header (SRH) is defined [RFC 8754](#) [[RFC8754](#)].

SRv6 Network Programming is defined  
[\[I-D.ietf-spring-srv6-network-programming\]](#).

The reader is expected to be familiar with the three above documents which define Segment Routing over the IPv6 data-plane (SRv6).

SRv6 is flexible and powerful, but in some uses cases, when a large number of SIDs are required, the size of the SRH/SID may be seen as too large both for some dataplane processors and traffic overhead.



The goal of this document is to propose a solution which reduces the size of the SRv6 header when a high number of SIDs is required with minimal changes to the SRv6 architecture, signaling, SRH and data plane processing.

This document extends SRH and SRv6 Network Programming to allow for SIDs of variable length, from 1 up to 128 bits. In a way, this is a generalization of SRv6 for any (v)SID length, where 128-bits SIDs is a special case.

### **1.1. Requirements Language**

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [BCP 14 RFC 2119](#) [[RFC2119](#)] [RFC 8174](#) [[RFC8174](#)] when, and only when, they appear in all capitals, as shown here.

## **2. Overview**

In a nutshell, SRv6 variable length SIDs (SRv6 vSIDs) proposes to:

- o define one SRv6 SIDs prefix dedicated to SRv6 vSIDs and called SRv6 vSIDs prefix;
- o define the vSID as the SRv6 SID minus the vSIDs prefix:  $\text{SRv6 SID} := \text{SRv6 vSIDs prefix} + \text{SRv6 vSID}$ ;
- o encode, in the Segment List of the SRH, only the vSIDs.

In other words, 128-bits SRv6 SIDs are kept unchanged for signaling, configuration and SR policies. In the data plane, this proposal compresses the encoding of SRv6 SIDs in the SRH by not encoding the SRv6 vSIDs prefixes which are redundant across vSIDs.

The use of the destination address in the IPv6 header is unchanged. The IPv6 destination address still contains the current/next 128-bits SRv6 SID. As a consequence, the IPv6 destination address already indicates the SRv6 vSID prefix and there is no extend the SRH to carry it.

In a way, this is similar to SR-MPLS [RFC 8660](#) [[RFC8660](#)]:

- o For SR-MPLS:  $\text{SR-MPLS Label} := \text{SRGB} + \text{Index}$
- o For SRv6 vSID:  $\text{SRv6 SID} := \text{SRv6 vSID prefix} + \text{SRv6 vSID}$



One difference compared to SR-MPLS is that lowest bits of the SRv6 vSID prefix are zero which allows for an easier operation in the data plane. Indeed, the addition of the vSID may be replaced by a copy of the vSID bits. Another difference is that the motivation to offset to a zero base index/vSID is different.

Except for the Segment List using (shorter) vSIDs, the format of the SRH is unchanged. The length of the vSIDs is defined to be 128 bits minus the length of the vSIDs prefix. It is variable but its length does not need to be encoded in the SRH header. Indeed the vSIDs length only needs to be known by the SR Segment Endpoint Node processing it. As per [section 4.3 of RFC 8754](#) [[RFC8754](#)], the SR node identifies its local SID by performing a longest-prefix-match lookup on the packet IPv6 destination address. This identifies the SID and its properties, such as the SR endpoint behavior but also the length of the vSIDs.

### 3. SRv6 Variable Length SIDs

An SRv6 vSID deployment choose one length 'L' of vSIDs and an associated SRv6 vSIDs prefix.

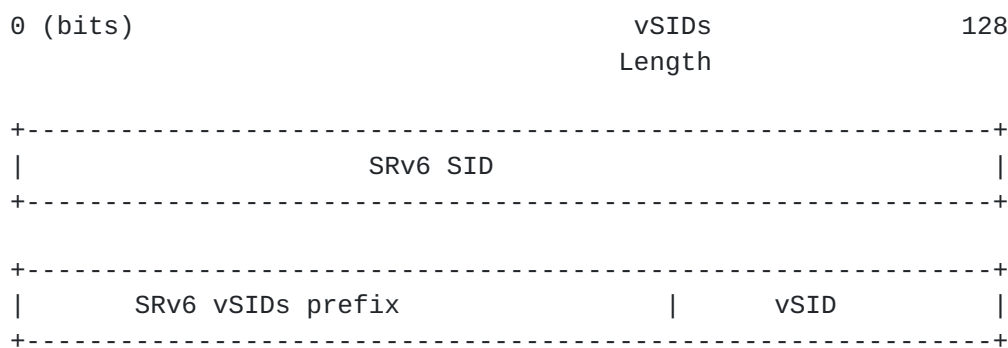


Figure 1: SRv6 SID:= SRv6 vSIDs prefix + SRv6 vSID

An SRv6 vSID deployment may use multiple SRv6 vSIDs prefixes. Each prefix may have its own vSIDs length.

As per section 3.1 of [[I-D.ietf-spring-srv6-network-programming](#)], an SRv6 SID can be represented as 'B:N:FUNCT:ARG'. Where 'B' is the SRv6 SID prefix, N is the identifier of the parent node N, FUNCT is the function of the SID of length 128-S.

For SRv6 vSIDs, the SRv6 prefix is 'B' of length LB as advertised in the SRv6 SID Structure Sub-Sub-TLV defined in [\[I-D.ietf-lsr-isis-srv6-extensions\]](#). The vSID length for vSIDs of this block is computed as 128 - LB. The vSID includes 'N:FUNCT:ARG'.



In order to simplify implementations, the length of the vSIDs MUST be a multiple of 8-bits, up to 128 bits included, in order to provide octet alignment in the SRH Segment List. However this specifications is written to work for any bit granularity from 1 to 128 bits. This allows for a future document to define a new profile with a higher granularity (e.g. 1 bit, 4 bits, 16 bits, or an integer fraction 128-bits) depending on hardware capabilities and flexibility requirements. To allow some implementations to be further simplified, a specific profile 'Profile-32' is created restricting the vSID length to be 32-bits.

### 3.1. Encoding vSIDs in the SRH

As per [section 2 of RFC 8754](#) [RFC8754], the Segment Routing Header (SRH) is defined as follows:

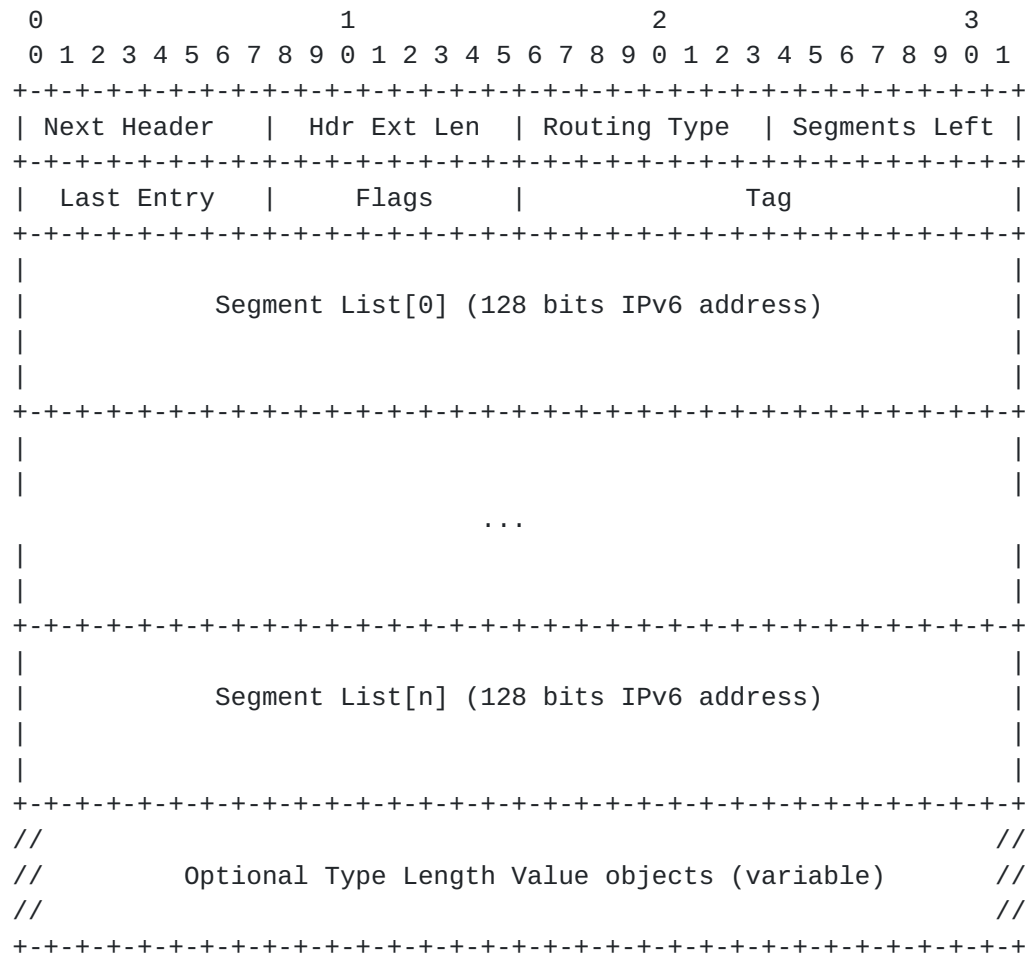


Figure 2: SRH with 128-bits SRv6 SID

When vSIDs are used, there are encoded in the Segment List of the Segment Routing Header (SRH).





Figure 3: SRH with SRv6 vSIDs with 16-bits vSIDs



### 3.2. SRv6 vSIDs Network Programming

This section defines a flavor for Endpoint behaviors. It defined below using the "End" Endpoint behavior but is applicable to all SID behaviors.

The processing takes as an argument the length L of the vSIDs. This length L is a property of the vSID and is given as a result of the lookup on the IPv6 destination address which identifies the SRv6 SID and its properties. The properties include the Endpoint behavior and the length L of the vSIDs.

When N receives a packet whose IPv6 DA is S and S is a local vSID of length L, the lines S08 and S14 of the End processing which were, as per section 4.1 of [[I-D.ietf-spring-srv6-network-programming](#)]:

```
S08.  max_LE = (Hdr Ext Len / 2) - 1
```

```
[...]
```

```
S14.  Update IPv6 DA with Segment List[Segments Left]
```

are replaced by the following:

```
S08.  max_LE = (Hdr Ext Len * 64 / L) - 1
```

```
[...]
```

```
S14.  Copy Segment List[Segments Left] from the SRH to the L lowest  
order bits of the destination address of the IPv6 header.
```

Note: S14. Taking into consideration that the Segment List is a list of vSIDs of length L bits

## 4. Benefits

SRv6 vSID is believed to have the following benefits:

- o Aligned with SRv6: SR architecture, SRv6 Network Programming.
- o Reduced SID hence reduced header length.
- o Flexible SID length, to accommodate for various deployment models, network sizes, SRv6 usages.
  - \* A typical vSIDs length could be 32 bits. Compared to SR-MPLS (which has a 20 bits SID) it is larger and more scalable.



Compared to SRv6 (which has a 128 bits SID) it's four times more compact.

- \* Other SID length are possible: 16 bits would be 8 times more compact than SRv6 SID and 2 times more compact the SR-MPLS shim header and large enough for most deployments; 8 bits would be 16 (respectively 4) more compact than SRv6 SID (respectively SR-MPLS shim header) and could fit some specific deployments (e.g. local adjacency SID only).
- o Using SRv6 header (SRH) with no additional fields.
- o No requirement for additional IPv6 addressing space: a /64 per router is more than enough. A /96 per router is the typical requirement.

## 5. Illustrations

This section illustrates the usage of SRv6 vSIDs through two examples. The first example uses global segments with a vSIDs locator been globally routable. The second example uses local segments only with a vSIDs locator common to all SR end nodes hence not been able to locate a specific SR end node. Global and local segments are defined as per [RFC 8402](#) [[RFC8402](#)] [section 2](#).

### 5.1. Global vSIDs

In this example vSIDs are used for global SIDs and are used alone without 128-bits SRv6 SIDs.

The SR domain has the following characteristics:

- o 1 000 SR endpoints nodes;
- o network diameter is 10;
- o vSIDs are chosen to be 32-bits long;
- o each SRv6 node is allocated a /108 to allocate its vSIDs from. This allows for 4 096 ( $2^{12}$ ) locators 1 million ( $2^{20}$ ) local functions on each SR node;
- o the SR domain and the SRv6 vSIDs prefix is allocated: 2001:DB8::/96;
- o node N is allocated 2001:DB8:0:0:0:0:N/108;



Some metrics of this SR domain:

- o An SR policy encoding a strictly routed path using only Adjacency SIDs would need ten 32-bits vSIDs resulting in a total of 40 octets in the SRH. In contrast the use of 128-bits SRv6 SIDs would require 160 octets;
- o An SR policy using strictly routed path using four (node) SIDs would need four 32-bits vSIDs resulting in a total of 16 octets in the SRH. In contrast the use of 128-bits SRv6 SIDs would require 64 octets and the use of 20-bits SR-MPLS SID would require 16 octets;
- o The IGP advertises 1 000 SRv6 locators to be installed in the IPv6 FIB of all IGP nodes (as per regular SRv6 and SR-MPLS);
- o The IPv6 address space is one /108 per SR node for a total of one /96 for the whole SR domain.

## **5.2. Local vSIDs**

If SRv6 vSIDs only identifies local segments, with no needs for global segments, there is no need for a global route toward a node hence the 'N' may have a size of zero. This may be interesting for a deployment using both 128-bits SRv6 SIDs and very short SRv6 vSIDs. Such SRv6 vSIDs could be used when a strictly routed path is needed and encoded as a list of adjacency SIDs. Given that the number of local adjacency SIDs is independent of the size of the SR domain, and typically below 255, one could use 8-bits vSIDs which would allow encoding 16 vSIDs within a single 128-bits SRv6 SID hence provides a very effective SRH compression.

In this example vSIDs are used only for local SIDs, such as adjacency SIDs. vSIDs are used in complement with 128-bits SRv6 SIDs.

The SR domain has the following characteristics:

- o 10 000 SR endpoints nodes;
- o network diameter is 30;
- o SRv6 SIDs:
  - \* each SRv6 node is allocated a /64 to allocate its 128-bits SIDs from;
  - \* SRv6 prefix: 2001:DB8::/48 (i.e., 65535 /64, allowing for growth or multiple SR routing algorithms);





- \* node N is allocated 2001:DB8:0:N/64;
- o SRv6 vSIDs
  - \* local vSIDs are chosen to be 8-bits in length. They are used for adjacency SIDs hence allow for 255 Adjacency SIDs per node;
  - \* SRv6 vSIDs prefix is allocated 2001:DB8:0:FFFF::/120;

Some metrics of this SR domain:

- o An SR policy encoding a strictly routed path using only adjacency SIDs would need thirty 8-bits vSIDs resulting in a total of 32 octets in the SRH. In contrast the use of 128-bits SRv6 SIDs would require 480 octets and the use of 20-bits SR-MPLS SID would require 120 octets;
- o The IGP advertises 10 000 SRv6 locators to be installed in the IPv6 FIB of all IGP nodes (as per regular SRv6 or SR-MPLS);
- o The IPv6 address space is one /64 per SR node for a total of one /48 for the whole SR domain.

## 6. Optional extensions

### 6.1. A node requiring a larger vSIDs length.

One SR Endpoint node may need a larger vSIDs space. This may especially be the case when at the same time:

- o the vSIDs Length is chosen to be small in order to optimize for the size of the SRH header. Indeed, for topological/routing instructions, the number of SIDs in the SRH may be high in some use cases, up to the network diameter, calling for small vSIDs.
- o one vSID is a service instruction and the number of service SIDs may be high, requiring a SID longer than a vSIDs Length.

This section defines two solutions to increase the vSIDs length on a node requiring a vSIDs space larger than other nodes.



### **6.1.1. Allocating a shorter locator to a node.**

In order to increase the vSIDs length on a node, a simple option is to allocate multiple locators of a larger locator to this node, within the SRv6 vSIDs prefix. As per [\[I-D.ietf-spring-srv6-network-programming\]](#), a locator may be represented as B:N where B is the SRv6 SID block (IPv6 subnet allocated for SRv6 SIDs by the operator) and N is the identifier of the parent node instantiating the SID. Following this format, multiple node identifiers Ns may be allocated to a node. Allocating 'k' identifiers increase the vSIDs space of that node by 'k'.

### **6.1.2. Combining k vSIDs to identify a behavior.**

When an SR Endpoint node needs more SIDs than allowed by the vSIDs Length, it MAY combine two (resp. N) vSIDs of length L to effectively benefit from a vSID of length 2\*L (resp. N\*L). This is a local choice of this SR Endpoint. Nothing specific is required in the SRH which only contains those 2 (resp. N) SIDs instead of a single one.

Note that when two vSIDs are combined, the first vSID may be seen as having the role of a "Context SID" identifying a context specific SID space/table, with the second SID been looked up in this context specific table. This is similar to the Context-Specific Label space defined in the [section 3 of RFC 5331 \[RFC5331\]](#).

This section defines the extension to SRv6 Network Programming allowing for a SID behavior "End.DT6" to be encoded with 'k' vSID. This is equally applicable to others SID behaviors.

The processing takes as an argument the vSIDs length 'L', and the number of vSIDs 'k'. 'L' and 'k' are a property of the vSID and are given as a result of the lookup of the IPv6 destination address which identifies the SRv6 SID and its properties. The properties include the Endpoint behavior, the length L of the vSIDs and the number 'k' of vSIDs used to encode this behavior in the SRH.

When N receives a packet destined to S and S is a local End.DT6 SID encoded with 'k' vSIDs, N does the following processing:

When N receives a packet whose IPv6 DA is S and S is a local End.DT6 vSID of length L and encoded with 'k' vSIDs the line S02 of the End.DT6 processing which was, as per section 4.6 of [\[I-D.ietf-spring-srv6-network-programming\]](#):

```
S02. If (Segments Left != 0) {
```



is replaced by the following:

S02. If (Segments Left  $\neq$  k - 1) {

When processing the Upper-layer header of a packet, the lines S04-S05 which were:

S04. Remove the outer IPv6 Header with all its extension headers

S05. Set the packet's associated FIB table to T

are replaced by the following:

S04a. Read the k-1 next vSIDs

S04b. Remove the outer IPv6 Header with all its extension headers

S05. Set the packet's associated FIB table to the one identified by the concatenation of the k-1 next vSIDs

## **6.2. Inter Routing Domains with the End.XvPS behavior**

Some SRv6 traffic may need to cross multiple routing domains, such as different Autonomous Systems (ASes) or different routing areas. Different routing domains may use different addressing schema making it difficult to find a common SRv6 vSIDs prefix.

This section defines an optional solution and SID behavior allowing for the use of different SRv6 vSIDs prefixes between routing domains.

The solution requires a new SID behavior, called "Endpoint with cross-connect to an array of layer-3 adjacencies and SRv6 vSIDs Prefix Swap" (End.XvPS for short) allowing for this transition of SRv6 vSIDs prefix between two routing domains.

End.XvPS is a variant of End.X, performing both "End.X Layer-3 Cross-Connect" and the translation of the SRv6 vSIDs prefix between the two routing domains.

The processing takes as an argument the vSIDs length L and the prefix P2 of the SRv6 vSIDs prefix in the second domain. Those two parameters are a property of the (received) vSID and are given as a result of the lookup on the IPv6 destination address which identifies the SRv6 SID and its properties.

When N receives a packet whose IPv6 DA is S and S is a End.XvPS SID behavior, with a local vSID length L, and a remote/next SRv6 vSIDs



prefix P2, the line S14 of the End processing which was, as per section 4.1 of [[I-D.ietf-spring-srv6-network-programming](#)]:

S14. Update IPv6 DA with Segment List[Segments Left]

is replaced by the following:

S14. Copy Segment List[Segments Left] from the SRH to the L lowest order bits of the destination address of the IPv6 header and copy the SRv6 vSIDs prefix P2 to the 128-L highest order bits of the destination address.

Note: the way the SRv6 vSIDs prefix P2 of the next routing domain is known is out of scope of this document. As examples, they could be learnt via configuration or using a signaling protocol either with the peer domain or with a central controller (e.g. PCE).

When End.XvPS SID behavior is used, the restriction on the SRH is relaxed and becomes: in a SRH, all vSID MUST have the same vSID length 'L' and within a routing domain, the same SRv6 vSID prefix. Between routing domains, separated with the End.XvPS behavior, SRv6 vSID prefixes may be different.

### **6.3. NSIDs flavor**

The NSIDs flavors is a variant of the End, End.X and End.XvPS behaviors.

The NSIDs flavors copy N vSID from the SRH to the IPv6 destination address.

When N receives a packet whose IPv6 DA is S and S is a NSIDs flavor, with a local vSID length L, and a local parameter N, the lines S09, S13 and S14 of the End processing which was, as per section 4.1 of [[I-D.ietf-spring-srv6-network-programming](#)]:

S09. If ((Last Entry > max\_LE) or (Segments Left > Last Entry+1)) {

S13. Decrement Segments Left by 1

S14. Update IPv6 DA with Segment List[Segments Left]

are replaced by the following:

S09. If ((Last Entry > max\_LE) or (Segments Left > Last Entry+1)) or (Segments Left <= N){

S13. Decrement Segments Left by N





S14. Copy N Segments List[Segments Left+N-1]...List[Segments Left] from the SRH to the N\*L lowest order bits of the destination address of the IPv6 header.

Note that if  $N=128/L$ , a complete IPv6 address is copied. This is useful when the next SR node is an SRv6 PE requiring its 128-bits SIDs in the IPv6 destination address. This is also useful for inter domain scenarios where both domains use unrelated address spaces and hence use two different SRv6 vSIDs prefixes.

## **7. Security Considerations**

This document does not change the security considerations of SRv6. Please refers to [RFC 8402](#) [[RFC8402](#)], [RFC 8754](#) [[RFC8754](#)] and [[I-D.ietf-spring-srv6-network-programming](#)] for existing security consideration.

## **8. Acknowledgements**

This document has been inspired by the work of the SPRING WG and in particular the work done in [[I-D.filsfils-spring-net-pgm-extension-srv6-usid](#)] and [[I-D.li-spring-compressed-srv6-np](#)]. The authors would like to acknowledge the authors of these two documents.

The authors would like to thank Joel Halpern and Francois Clad for their review and comments.

## **9. Contributors**

Daniel Voyer

Bell Canada

Canada

Email: [daniel.voyer@bell.ca](mailto:daniel.voyer@bell.ca)

## **10. Changes / Authors Notes**

[RFC Editor: Please remove this section before publication]

00: Initial version.

01:

- o Removal of the vSID Size TLV; addition of the IS-IS extension; addition of the SR header length check in the pseudo code.



- o Addition of the IS-IS extension;
- o Addition of the SR header length check in the pseudo code.

02:

- o Change of terminology (VLSID changed to vSID);
- o Swapping the two examples in section 'Illustrations'.;
- o Addition of two solutions to provide more vSID space to some nodes in section 'Node requiring a larger vSID length';
- o Addition of a solution for crossing routing domains using different SRv6 vSID prefixes and/or different vSID length;

03:

- o Minor updates in section 'Inter Routing Domains'

04:

- o Creating of a profile only supporting vSID of length 32-bits.
- o Addition of the NSIDs flavor.

05:

- o Editorial changes.
- o Removal of the IGP signaling extension. The size of the SRv6 block is advertised in the existing SRv6 SID Structure Sub-Sub-TLV.

## **11. References**

### **11.1. Normative References**

[I-D.ietf-lsr-isis-srv6-extensions]

Psenak, P., Filsfils, C., Bashandy, A., Decraene, B., and Z. Hu, "IS-IS Extension to Support Segment Routing over IPv6 Dataplane", [draft-ietf-lsr-isis-srv6-extensions-11](#) (work in progress), October 2020.



- [I-D.ietf-spring-srv6-network-programming]  
Filsfils, C., Camarillo, P., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "SRv6 Network Programming", [draft-ietf-spring-srv6-network-programming-28](#) (work in progress), December 2020.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", [RFC 8402](#), DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.
- [RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", [RFC 8754](#), DOI 10.17487/RFC8754, March 2020, <<https://www.rfc-editor.org/info/rfc8754>>.

## **11.2. Informative References**

- [I-D.filsfils-spring-net-pgm-extension-srv6-usid]  
Filsfils, C., Camarillo, P., Cai, D., Voyer, D., Meilik, I., Patel, K., Henderickx, W., Jonnalagadda, P., Melman, D., Liu, Y., and J. Guichard, "Network Programming extension: SRv6 uSID instruction", [draft-filsfils-spring-net-pgm-extension-srv6-usid-08](#) (work in progress), November 2020.
- [I-D.li-spring-compressed-srv6-np]  
Li, Z., Li, C., Xie, C., LEE, K., Tian, H., Zhao, F., Guichard, J., Cong, L., and S. Peng, "Compressed SRv6 Network Programming", [draft-li-spring-compressed-srv6-np-02](#) (work in progress), February 2020.
- [RFC5331] Aggarwal, R., Rekhter, Y., and E. Rosen, "MPLS Upstream Label Assignment and Context-Specific Label Space", [RFC 5331](#), DOI 10.17487/RFC5331, August 2008, <<https://www.rfc-editor.org/info/rfc5331>>.



[RFC8660] Bashandy, A., Ed., Filsfils, C., Ed., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with the MPLS Data Plane", [RFC 8660](#), DOI 10.17487/RFC8660, December 2019, <<https://www.rfc-editor.org/info/rfc8660>>.

#### Authors' Addresses

Bruno Decraene  
Orange

Email: [bruno.decraene@orange.com](mailto:bruno.decraene@orange.com)

Robert Raszuk  
NTT Network Innovations  
940 Stewart Dr  
Sunnyvale, CA 94085  
USA

Email: [robert@raszuk.net](mailto:robert@raszuk.net)

Zhenbin Li  
Huawei Technologies

Email: [lizhenbin@huawei.com](mailto:lizhenbin@huawei.com)

Cheng Li  
Huawei Technologies

Email: [chengli13@huawei.com](mailto:chengli13@huawei.com)



